ENCODING-BASED MULTICAST PACKET DUPLICATION CONTROL SUITABLE FOR VLAN SYSTEMS

Inventors:

Gaurav Singh
Govind Malalur
Brian Hang Wai Yang

FIELD

[0001] The invention relates generally to the field of networking, and more particularly to packet duplication control.

BACKGROUND

[0002] In networking systems, routers and/or switches typically move packets of information from one of a number of input ports to one or more output ports. In many applications, such as multicasting, a packet may need to be duplicated on multiple output ports. Further, for support of Virtual Local Area Network (VLAN or "Vlan") systems, a packet may need to be duplicated according to a wide variety of possibilities. In a typical Vlan system, 12-bits may be used to indicate a particular Vlan, so the number of possible distinct Vlans that need to be supported in the system can be as high as 4096. These requirements pose a challenge to architect the multicast packet duplication logic for Vlan systems where up to about 4K distinct Vlans can be supported per port while also maintaining the minimum duplication requirements.

[0003] A block diagram of a conventional router/switch port arrangement used for packet duplication for Vlan systems is shown in FIG. 1 and indicated by the general reference character 100. In FIG. 1, an Internet Protocol MultCast (IPMC) type packet is received by Router/Switch 102 at input port or interface X. The output ports or interfaces include port 0 requiring 3 duplicates of the packet, which are designated for Vlan2, Vlan8, and Vlan9. Output port 1 in this example only requires one duplicate for Vlan5. Output port N requires 2 duplicates: one for Vlan4 and one for Vlan7. The duplication possibilities could range from 2K times on one interface, 3K on another, 3K on another, or just one on each, for example. Further, each copy of the multicast packet can be sent to a different Vlan on a different port or multiple copies might need to be sent on each port. Accordingly, a hardware solution must be

flexible enough to accommodate IPMC or other similar type packets requiring duplication on a variety of designated member ports.

[0004] Such a wide disparity of options is typically implemented in conventional approaches by simply expanding the memory used to cover all the possible cases, with little flexibility. However, such an inefficient use of memory is not desirable due to cost factors.

[0005] Consequently, what is needed is a solution that can control packet duplication in a flexible and memory efficient manner.

SUMMARY

[0006] The invention provides a structure and method for controlling packet duplication in a flexible and memory efficient manner.

[0007] According to embodiments of the invention, a packet duplication system includes an input port for receiving a packet and a plurality of output ports for outputting duplications of the packet. The duplications can be suitable to support a Virtual Local Area Network (VLAN) system. The duplications can be controlled by descriptors arranged in a linked-list table. Also, the descriptors can have encoding formats, such as contiguous range encoding, non-contiguous range encoding, and discrete encoding. Further, the linked-list table can include at least one shared descriptor.

[0008] According to another aspect of the invention, a method of controlling a duplication of a packet includes receiving the packet, accessing a first pointer, accessing a second pointer, accessing a descriptor in response to the second pointer, and applying an encoding format for the duplication of the packet. The encoding format can include contiguous range encoding, non-contiguous range encoding, and/or discrete encoding. Further, the method can include the use of a hashing type function. The method can also be suitable to support a Virtual Local Area Network (VLAN) system.

[0009] The invention can provide a fully configurable and flexible duplication scheme. This approach allows for efficient use of on-chip memory for systems requiring a wide range of duplication options over multiple ports.

BRIEF DESCRIPTION OF THE FIGURES

[0010] Embodiments of the invention are described with reference to the FIGS, in which:

[0011] FIG. 1 a block diagram of a conventional router/switch port arrangement used for packet duplication for Vlan systems.

[0012] FIG. 2 is a block diagram of a multicast forwarding system according to an embodiment.

[0013] FIG. 3 is a block diagram of a multicast Vlan linked-list structure according to an embodiment.

[0014] FIG. 4A is a diagram of a multicast Vlan linked-list descriptor structure according to an embodiment.

[0015] FIG. 4B is a diagram of a contiguous Vlan range sub-descriptor encoding scheme structure according to an embodiment.

[0016] FIG. 4C is a diagram of a non-contiguous Vlan range sub-descriptor encoding scheme structure according to an embodiment.

[0017] FIG. 4D is a diagram of a discrete Vlan encoding scheme structure according to an embodiment.

[0018] FIG. 5 is a flow diagram of a packet duplication control method suitable for Vlan systems according to an embodiment.

DETAILED DESCRIPTION

[0019] Embodiments of the invention are described with reference to specific diagrams depicting system arrangements and methods. Those skilled in the art will recognize that the description is for illustration and to provide the best mode of practicing the invention. The description is not meant to be limiting. For example, reference is made to several types of packets and application systems, such as Virtual Local Area Network (VLAN or "Vlan") systems, but the invention is applicable to other types of packets and/or systems as well. Further, pointer and/or descriptor table organization, including separate tables and the like, merely provide example implementations and should not be construed as limiting.

[0020] Referring now to FIG. 2, a block diagram of a multicast forwarding system according to an embodiment is shown and indicated by the general reference character 200. A MultiCast (MC)Address can be received and passed through Hashing Function 202 to index Multicast Table 204. Multicast Table 204 can include a plurality of pointers, such as shown Pointer 204-0, Pointer 204-X, and Pointer 204-M. For example, the result of Hashing Function 202 can select Pointer 204-X including MulticastDescriptor 206. MulticastDescriptor 206 can include several fields, such as field 206-0, which may be a Time-To-Live (TTL) type field,

VlanPtr 206-1, field 206-2, and field 206-3. VlanPtr 206-1 can, for example, select Pointer 208-X of McVlan (Multicast Vlan) Table 208. McVlan Table 208 can include multiple pointers, such as ranging from Pointer 208-0 through 208-M. Each of the pointers in McVlan Table 208 can have a width of entries where each entry corresponds to an output (i.e., exit) port of the system. This can allow the packet duplication for Vlans to essentially be done on a per-port basis. Pointer 208-X can include VlanPtrDescriptor 210, which can include several linked-list pointers corresponding to system output ports, such as VlanLLPtrA 210-A and VlanLLPtrB 210-B through VlanLLPtrH 210-H. In this example, VlanLLPtrA can select Descriptor 212-0 from VlanLinkList Table 212. Descriptor 212-0 can include a link to Descriptor 212-P, which can include a link to Descriptor 212-S, and the chain can continue if necessary. In this example case, there are 8 ports of exit, each corresponding to one of the pointers chosen from VlanLLPtrA through VlanLLPtrH. Accordingly, VlanLLPtrA can be an index into the VlanLinkList Table 212 for Port "A," VlanLLPtrB can be an index for Port "B," and so on.

[0021] Referring now to FIG. 3, a block diagram of a multicast Vlan linked-list structure according to an embodiment is shown and indicated by the general reference character 300. This structure is only one example of the many possible link structures available according to embodiments of the linked-list table discussed above with reference to FIG. 2. In FIG. 3, VlanLLPtrA may point to VlanLLDescriptor 302, VlanLLPtrB may point to VlanLLDescriptor 304, VlanLLPtrC may point to VlanLLDescriptor 306, VlanLLPtrD may point to VlanLLDescriptor 308, VlanLLPtrE may point to VlanLLDescriptor 310, and VlanLLPtrF may point to VlanLLDescriptor 312. In addition, the structure allows for sharing descriptors, such as those shown in the SharedVlanLLDescriptors dashed box. VlanDescriptor 314 can be linked by VlanLLDescriptor 302 and VlanLLDescriptor 304 and it can point to VlanLLDescriptor 320. VlanLLDescriptor 316 can be linked by VlanLLDescriptor 306, VlanLLDescriptor 308, and VlanLLPtrG. Further, VlanLLDescriptor 316 can point to VlanLLDescriptor 320. VlanLLDescriptor 318 can be linked by VlanLLDescriptor 310 and VlanLLDescriptor 312 and it can point to VlanLLDescriptor 320. Further, VlanLLPtrH can also point to VlanLLDescriptor 320. As indicated, this illustration shows merely one possible linking structure, but any suitable structure can be formed using the flexible linking options according to embodiments. Further, a system design can be optimized so that the available storage space is allocated for a typical case while still providing support for up to 4K distinct

Vlans on each port. The sharing of multiple linked-lists, in particular, can significantly increase the efficiency of use of the available storage and/or table space.

[0022] Next, a few example encoding formats will be described. Such formats can allow for optimized use of the available table space depending on the particular distribution of the Vlans in the system. Of course, other possible encoding formats can also be used according to embodiments.

[0023] Referring now to FIG. 4A, a diagram of a multicast Vlan linked-list descriptor structure according to an embodiment is shown and indicated by the general reference character 400. The descriptor describes the Vlans that a packet is to be multicast on and several possible formats can be used. One example format is shown in FIG. 4A and it includes Format 402, which can be 8-bits or less, SubDescriptor1 404-1, SubDescriptor2 404-2, SubDescriptor3 404-3, SubDescriptor4 404-4, and NxtPtr 406. Format 402 can include control bits for use in selecting a format for each of the sub-descriptors in that descriptor, for example. Also, each of the sub-descriptors can be 24-bits wide, as one example. Further, while four sub-descriptors are shown per descriptor, more or less than four sub-descriptors can be employed, depending on the optimal system and/or memory arrangement. Further, sub-descriptors can be linked together, as discussed above with reference to FIGS. 2 and 3, to form larger linked structures. The number of bits available for the NxtPtr field depends on system design considerations, such as the size of the linked-list memory allocated and the number of sub-descriptors chosen per descriptor.

[0024] Referring now to FIG. 4B, a diagram of a contiguous Vlan range sub-descriptor encoding scheme structure according to an embodiment is shown and indicated by the general reference character 420. In this format, StartingVlan 422 can use 12-bits to indicate the beginning of the contiguous range and EndingVlan 424 can use the remaining 12-bits to indicate the end of the contiguous range. Accordingly, an associated packet can be duplicated on all the Vlans designated in the range.

[0025] Referring now to FIG. 4C, a diagram of a non-contiguous Vlan range sub-descriptor encoding scheme structure according to an embodiment is shown and indicated by the general reference character 440. In this format, VlanMSB 442 can use 8-bits to indicate a most significant bit portion and Bitmap 444 can indicate a pattern using the least significant bits. -The Vlan range can include Vlans that share the same 8-bits of MSB but with different 4-bit LSB portions. The lower 4-bits of the LSB for each Vlan in this range can be decoded into a 16-bit vector. The 16-bit vectors derived from the lower 4-bits of all the Vlans in the range

may be logically OR-ed to generate the final 16-bit - bitmap value. Each bit of this bitmap can correspond to a Vlan for which a packet should be duplicated. Accordingly, an associated packet can be duplicated on all the Vlans as designated in the range.

[0026] Referring now to FIG. 4D, a diagram of a discrete Vlan encoding scheme structure according to an embodiment is shown and indicated by the general reference character 460. In this format, Vlan1 462 can use 12-bits to indicate a first Vlan and Vlan2 464 can use the remaining 12-bits to indicate a second Vlan. Accordingly, an associated packet can be duplicated on the designated Vlans.

[0027] Referring now to FIG. 5, a flow diagram of a packet duplication control method suitable for Vlan systems according to an embodiment is shown and indicated by the general reference character 500. A packet can be received in step Receive Packet 502. A hashing function can be performed in step Perform Hashing 504, but this step may not be necessary depending on the size and type of the available lookup tables. A first pointer can be obtained in step Access VlanPtr 506 and that pointer can designate a second pointer in step Access VlanLLPtr 508. The second pointer, as part of a linked-list structure, can designate a descriptor in step Access VlanLLDescriptor 510. Once a descriptor is found, a format determination for each of its sub-descriptors can begin, for example, with step Determine SubDescriptor Encoding Format 512. Of course, such encoding determination may also be implemented in a more parallel fashion. If decision branch Contiguous Range 514 is true, the flow can proceed to Apply Contiguous Vlan Range Encoding 518 and then to decision branch Another Descriptor Link 524. If decision branch Contiguous Range 514 is false, the flow can proceed to decision branch Non-Contiguous Range 516 which, if true, the flow can proceed to Apply Non-Contiguous Vlan Range Encoding 520 and then to decision branch Another Descriptor Link 524. If decision branch Non-Contiguous Range 516 is false, the flow can proceed to Apply Discrete Vlan Encoding 522 and then to decision branch Another Descriptor Link 524. If decision branch Another Descriptor Link 524 is true, the flow can proceed back to step Access VlanLLDescriptor 510. Otherwise, the flow can proceed back to step Receive Packet 502. In this fashion, duplication of packets can be controlled according to bit encoding interpretations found within linked data structures.

[0028] Advantages of the invention include the control of packet duplication over a variety of output ports in a memory efficient and flexible manner.

[0029] Having disclosed exemplary embodiments and the best mode, modifications and variations may be made to the disclosed embodiments while remaining within the subject and spirit of the invention as defined by the following claims.